

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:)
Yutaka HAGA)
Serial No.: To be assigned) Group Art Unit: Unassigned
Filed: February 6, 2001) Examiner: Unassigned
For: APPARATUS FOR COLLECTING) PROFILES OF PROGRAMS)

JC918 U.S. PTO
09/778076
02/07/01

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. §1.55**

*Assistant Commissioner for Patents
Washington, D.C. 20231*

Sir:

In accordance with the provisions of 37 C.F.R. §1.55, the applicants submit herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-292736
Filed: September 26, 2000.

It is respectfully requested that the applicants be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,
STAAS & HALSEY LLP

Date: February 6, 2001

By: _____

James D. Halsey, Jr.
Registration No. 22,729

700 11th Street, N.W., Ste. 500
Washington, D.C. 20001
(202) 434-1500

CERTIFIED COPY OF
PRIORITY DOCUMENT

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

0p113t



別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出 願 年 月 日
Date of Application:

2000年 9月26日

出 願 番 号
Application Number:

特願2000-292736

出 願 人
Applicant (s):

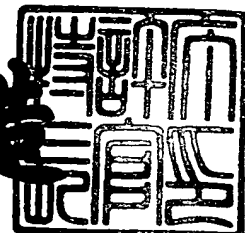
富士通株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年11月10日

特許庁長官
Commissioner,
Patent Office

及川耕造





【書類名】 特許願

【整理番号】 0050858

【提出日】 平成12年 9月26日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 11/34

【発明の名称】 プログラム性能情報収集装置及びコンピュータ読取可能な記録媒体

【請求項の数】 5

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 芳賀 豊

【特許出願人】

 【識別番号】 000005223

 【氏名又は名称】 富士通株式会社

【代理人】

 【識別番号】 100089244

 【弁理士】

 【氏名又は名称】 遠山 勉

【選任した代理人】

 【識別番号】 100090516

 【弁理士】

 【氏名又は名称】 松倉 秀実

 【連絡先】 03-3669-6571

【手数料の表示】

 【予納台帳番号】 012092

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705606

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 プログラム性能情報収集装置及びコンピュータ読取可能な記録媒体

【特許請求の範囲】

【請求項 1】 プログラムに含まれたサブルーチンの性能情報を収集する装置であって、

前記プログラムの実行中に分岐命令が実行されたときに発生する割込によって、前記サブルーチンの性能情報を収集する、プログラム性能情報収集装置。

【請求項 2】 サブルーチンの実行体毎に前記性能情報を収集する、請求項 1 記載のプログラム性能情報収集装置。

【請求項 3】 特定の実行体が複数のサブルーチンを実行する場合に、前記各サブルーチンの性能情報を個別に収集する、請求項 2 記載のプログラム性能情報収集装置。

【請求項 4】 或るサブルーチンについて、メインルーチンから呼び出された場合の性能情報と、他のサブルーチンから呼び出された場合の性能情報とを個別に収集する、請求項 3 記載のプログラム性能情報収集装置。

【請求項 5】 分析対象のプログラムに含まれたサブルーチンの性能情報を収集する処理をコンピュータに実行させるプログラムを記録した記録媒体であって

コンピュータに、

前記プログラムの実行中に分岐命令が実行されたことを契機として割込が発生した場合に、実行された分岐命令がサブルーチンの実行に関する命令か否かを判別するステップと、

前記分岐命令がサブルーチンの実行に関する命令と判別された場合に、前記サブルーチンの性能情報を取得して性能情報記憶部に格納するステップと、を実行させるプログラムを記録したコンピュータ読取可能な記録媒体。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は、コンピュータシステム上で実行されるプログラムの性能情報(プロフィール)を収集するプログラム性能情報収集装置、及びコンピュータをプログラム性能情報収集装置として機能させるプログラムを記録したコンピュータ読取可能な記録媒体に関する。

【 0 0 0 2 】

【従来の技術】

従来、サブルーチン(「手続」,「関数」等とも呼ばれる)の詳細な性能分析(プロファイリング)を行なう方法として、以下の方法がある。

(1)サブルーチンのプロファイリング用の制御コードを予めプログラム中に埋め込む。

(2)プログラムを解析し、サブルーチンの呼出命令を横取りするようにプログラムを改変する。

【 0 0 0 3 】

【発明が解決しようとする課題】

しかしながら、上記した従来の方法では以下の問題があった。即ち、上記(1)に示したプロファイリング用の制御コードをプログラム中に埋め込む方法では、プログラムの実行時におけるオーバーヘッドが大きくなる。このため、(1)の方法は、開発段階でのプログラムに適用され、埋め込まれた制御コードは、出荷される最終製品のプログラムから省かれることが多かった。従って、(1)の方法を最終製品のプログラムに適用することは困難であった。

【 0 0 0 4 】

また、上記(2)に示した呼出命令を横取りするようにする方法では、プログラムコードを改変する。このため、プログラム自身の正当性検査で改変が検出された場合に、検出された改変が不当と判定され、プログラムが動作しなくなる可能性があった。

【 0 0 0 5 】

本発明の目的は、プログラムに改変を加えることなくサブルーチンの性能情報を収集することができるプログラム性能情報収集装置及びコンピュータ読取可能な記録媒体を提供することである。

【 0 0 0 6 】

【課題を解決するための手段】

本発明は、上述した目的を達成するために以下の構成を採用する。即ち、本発明は、プログラムに含まれたサブルーチンの性能情報を収集する装置であって、前記プログラムの実行中に分岐命令が実行されたときに発生する割込によって、前記サブルーチンの性能情報を収集する。

【 0 0 0 7 】

本発明によれば、割込によって性能情報(プロフィール)を収集するので、プログラムに制御コードを埋め込んだり、改変を加えたりする必要がない。従って、例えば、最終製品のプログラム中のサブルーチンの性能情報を収集し、分析することが可能となる。性能情報は、例えば、サブルーチンの累計実行時間、呼出回数、オーバーヘッド等である。

【 0 0 0 8 】

本発明は、サブルーチンの実行体毎に前記性能情報を収集する構成としても良い。このようにすれば、詳細なサブルーチンの性能情報を収集することができ、詳細なプログラムの分析が可能となる。実行体は、プロセス(「タスク」とも呼ばれる)或いはスレッドである。

【 0 0 0 9 】

本発明は、さらに、特定の実行体が複数のサブルーチンを実行する場合に、前記各サブルーチンの性能情報を個別に収集する構成としても良い。このようにすれば、さらに詳細なサブルーチンの性能情報を収集することができ、詳細なプログラムの分析を行うことが可能となる。

【 0 0 1 0 】

本発明は、さらに、或るサブルーチンについて、メインルーチンから呼び出された場合の性能情報と、他のサブルーチンから呼び出された場合の性能情報とを個別に収集する構成としても良い。このようにすれば、或るサブルーチンについて、メインルーチンから呼び出された場合の性能情報と、他のサブルーチンから呼び出された(ネスティングによって呼び出された)場合の性能情報とを、個別に収集することができ、詳細なプログラムの分析が可能となる。

【 0 0 1 1 】

本発明は、さらに、他のサブルーチンから呼び出された場合の性能情報を、他のサブルーチンと呼び出されたサブルーチンとの関係を示す呼出関係情報と関連づけて保持する構成としても良い。このようにすれば、各サブルーチン間の呼出関係(ネスティングの状態)を含む詳細な性能情報を得ることができ、詳細なプログラムの分析が可能となる。

【 0 0 1 2 】

また、本発明は、分析対象のプログラムに含まれたサブルーチンの性能情報を収集する処理をコンピュータに実行させるプログラムを記録した記録媒体であって、コンピュータに、前記プログラムの実行中に分岐命令が実行されたことを契機として割込が発生した場合に、実行された分岐命令がサブルーチンの実行に関する命令か否かを判別するステップと、前記分岐命令がサブルーチンの実行に関する命令と判別された場合に、前記サブルーチンの性能情報を取得して性能情報記憶部に格納するステップと、を実行させるプログラムを記録したコンピュータ読取可能な記録媒体である。

【 0 0 1 3 】

【発明の実施の形態】

以下、図面を参照して本発明の実施形態を説明する。

【 0 0 1 4 】

〔第 1 実施形態〕

〈ハードウェア構成〉

図 1 は、本発明の実施形態によるプログラム性能情報収集装置として機能するコンピュータ 1 のハードウェアブロック図である。コンピュータ 1 は、例えば、パーソナルコンピュータ(PC)、ワークステーション(WS)を用いて構成される。コンピュータ 1 は、CPU 2、メインメモリ(MM) 3、二次記憶装置 4、キーボード 5、ポインティングデバイス 6 及びディスプレイ 7 が相互に接続されてなる。

【 0 0 1 5 】

二次記憶装置 4 は、本発明によるコンピュータ読取可能な記録媒体に相当し、

半導体メモリ(例えば、ROM, RAM, SRAM, フラッシュメモリ, EPROM, EEPROM), 磁気ディスク(例えば、ハードディスク, フロッピーディスク), 光ディスク(例えば、CD-ROM, PD), 光磁気ディスク(MO)等の各種の記録媒体を用いて構成される。二次記憶装置4は、オペレーティングシステム(OS)8と、少なくとも1つのアプリケーションプログラム9(以下、「アプリケーション9」という)と、性能分析用のプログラム10(以下、「分析プログラム10」という)がインストールされている。また、二次記憶装置4は、各分析プログラム8, 9, 10の実行時に使用されるデータを保持している。

【0016】

OS8は、コンピュータ1のハードウェア(CPU2, MM3等)とアプリケーション9との間に介在し、ハードウェアの違いを吸収する。また、ハードウェアの実行制御や管理を行う。OS8は、アプリケーション9を実行するための少なくとも1つの実行体19を生成する(図2参照)。図2には、例として、3つの実行体19(19A, 19B, 19C)が示されている。

【0017】

アプリケーション9は、OS8上で実行される応用プログラムであり、本発明における分析対象のプログラムに相当する。アプリケーション9は、メインルーチン及び単数又は複数のサブルーチンとの組み合わせ、或いは、複数のサブルーチンの組み合わせで構成される。この例では、アプリケーション9は、メインルーチンと複数のサブルーチンとの組み合わせで構成されている。

【0018】

分析プログラム10は、アプリケーション9のサブルーチンに関する性能情報(プロファイル)を収集するためのプログラム、即ち、コンピュータ1を本発明のプログラム性能情報収集装置として機能させるためのプログラムである。

【0019】

分析プログラム10は、CD-ROMやフロッピーディスク等の可搬性を有する記録媒体から二次記憶装置4(例えば、ハードディスク)にインストールするようにしても良く、コンピュータ1が分析プログラム10を通信回線を通じて他のコンピュータからダウンロードし、二次記憶装置4にインストールするようにし

ても良い。

【 0 0 2 0 】

キーボード 5 及びポインティングデバイス 6 は、コンピュータ 1 にコマンドやデータを入力するために使用される。以下、キーボード 5 とポインティングデバイス 6 とをまとめて指す場合には、「入力装置 1 1」と表記する。ポインティングデバイス 6 は、例えば、マウス、ジョイスティック、トラックボール、フラットスペースである。

【 0 0 2 1 】

ディスプレイ 7 は、陰極線管 (C R T)、液晶ディスプレイ (L C D)、プラズマディスプレイ等を用いて構成されており、ユーザがデータやコマンドを入力するための情報や、プログラムの実行結果等を表示する。

【 0 0 2 2 】

MM 3 は、C P U 2 の作業領域として使用される。また、MM 3 は、ディスプレイ 7 に情報を表示するためのデータを保持するビデオメモリ (V R A M) としても機能する。

【 0 0 2 3 】

C P U 2 は、二次記憶装置 4 に保持された O S 8、アプリケーション 9、分析プログラム 1 0 を MM 3 にロードして実行する。C P U 2 が O S 8 上でアプリケーション 9 を実行しているときに分析プログラム 1 0 を実行することにより、コンピュータ 1 は、本発明のプログラム性能情報収集装置として機能する。

【 0 0 2 4 】

〈機能的構成〉

図 2 は、図 1 に示したコンピュータ 1 によって実行されるプログラム性能情報収集装置の機能ブロック図である。図 2 (A)、(B) に示すように、プログラム 1 0 が C P U 2 によって実行されると、実行環境設定部 1 5 と、命令分析部 1 6 と、性能情報収集部 1 7 とが実現され、少なくとも 1 つの制御表 1 8 が作成される。以下、各部 1 5、1 6、1 7 の機能を説明する。

【 0 0 2 5 】

(実行環境設定部)

図 2 (A)に示すように、プログラム 1 0 が CPU 2 によって実行されると、実行環境設定部 1 5 が起動する。実行環境設定部 1 5 は、入力装置 1 1 を通じたユーザからの操作に従って、アプリケーション 9 の性能情報を収集するための環境設定を CPU 2 に施す。

【 0 0 2 6 】

現在コンピュータ 1 に適用されるあらゆる CPU 2 は、プログラムの実行に際して割込を発生させる機能を有している。PC 等で良く使用されている X 8 6 アーキテクチャを有する CPU は、1 つの命令の実行時にシングルステップ割込を発生させることができる。また、分岐命令が実行された場合にのみ割込を発生させる機能をさらに有する CPU もある。

【 0 0 2 7 】

上記機能を有する CPU として、例えば、Intel 社の “Pentium Pro” を挙げることができる。また、“Pentium Pro” 以降の “Pentium II” や “Pentium III” も同様の機能を有している。

【 0 0 2 8 】

“Pentium Pro” は、EFLAGS レジスタと呼ばれる各種のシステム・フラグを含むレジスタを有している。この EFLAGS レジスタの 8 番目のビット(ビット 8)は、“TF フラグ” と呼ばれている。TF フラグがセットされると、デバッグのためのシングル・ステップ・モードがイネーブルとなる。逆に、TF フラグがクリアされると、デバッグのためのシングル・ステップ・モードがディスエーブルとなる。

【 0 0 2 9 】

シングル・ステップ・モードでは、プロセッサ(CPU)は、各命令の後にデバッグ例外を生成する。このため、各命令後にプログラムの実行状態を調べることができる。アプリケーションプログラムが POPF, POPFD, または IRET 命令を使用して TF フラグをセットすると、POPF, POPFD, または IRET 命令の後に続く命令後に、デバッグ例外が発生する。

【 0 0 3 0 】

また、“Pentium Pro” は、TF フラグを用いたデバッグ機能よりも詳細なデ

バグ機能を設定するレジスタとして、DebugCtlMSRレジスタを有している。DebugCtlMSRレジスタは、最新の分岐、割込、及び例外の記録をイネーブルにする。また、実行された分岐に対するブレイクポイント、ブレイクポイント報告ピン、及びトレースメッセージをイネーブルにする。

【 0 0 3 1 】

DebugCtlMSRレジスタには、特権レベル0での動作時又は実アドレスモード時にWRMSR命令を使用して書くことができる。DebugCtlMSRレジスタへのユーザアクセスを可能にするには、保護モードのオペレーティング・システム・プロシージャを要する。

【 0 0 3 2 】

DebugCtlMSRレジスタは、LBR(最新分岐／割込／例外)フラグ(ビット0)と、BTF(分岐シングル・ステップ実行)フラグ(ビット1)とを含んでおり、本発明はこれらのフラグを利用する。LBRフラグがセットされると、プロセッサ(CPU)はデバッグ例外が発生する前に最後に行った分岐、及び例外又は割込処理のソースアドレスとターゲットアドレスとを記録する。

【 0 0 3 3 】

一方、BTFフラグがセットされると、プロセッサ(CPU)は、EFLAGSレジスタのTFフラグを“命令シングル・ステップ実行”フラグとしてではなく、“分岐シングル・ステップ実行”フラグとして取り扱う。LBRフラグ及びBTFフラグの適宜の利用により、分岐処理をCPUにシングル・ステップ実行させることが可能となる。分岐に対するデバッグ・ブレイクポイントをイネーブルするには、実行環境設定部15は、BTFフラグ及びTFフラグの双方をセットしなければならない。

【 0 0 3 4 】

実行環境設定部15は、上述したLBRフラグ、BTFフラグ及びTFフラグを組み合わせ設定することにより、分岐命令実行時にのみ割込を発生させ、その時の分岐元アドレス及び分岐先アドレスを取得できるようにする。

【 0 0 3 5 】

即ち、実行環境設定部15は、被分析対象のプログラム(アプリケーション9)

がOS 8上で実行するための実行体19が生成されるときに、その実行体19の実行コンテキストを設定するとともに、CPU 2の分岐命令実行時における割込発生機能を有効化する。

【0036】

なお、実行環境設定部15は、自動的に上記環境設定を実行するように構成されていても良い。また、実行環境設定部15以外の方法で上記設定をCPU 2に施すことができる場合には、分析プログラム10によって実行環境設定部15が実現されるようにしなくても良い。

【0037】

(命令分析部)

図2(B)に示すように、アプリケーション9の実行時に分析プログラム10が実行されると、命令分析部16と、性能情報収集部17とが実現される。命令分析部16は、割込が発生した場合にCPU 2から制御を受け取るように、CPU 2の割込ハンドラとして構成される。

【0038】

命令分析部16は、割込の発生によってCPU 2から制御を受け取る場合に、実行されたアプリケーション9中の分岐命令の分岐元アドレス及び分岐先アドレスをCPU 2から受け取る。

【0039】

命令分析部16は、分岐元アドレス及び分岐先アドレスを取得すると、分岐元アドレスの命令コードをMM 3から読み出し、読み出した命令コードをデコードすることによって分岐命令の種別を判別する。分岐命令の種別は、サブルーチンの呼出命令、復帰命令、ジャンプ命令からなる。

【0040】

命令分析部16は、命令種別が呼出命令又は復帰命令である場合には、分岐元アドレス、分岐先アドレス及び分岐命令種別(呼出命令又は復帰命令)を性能情報収集部17に与える。

【0041】

なお、CPU 2によっては、一旦割込が発生すると、実行コンテキストから再

び割込が発生しないように、制御レジスタの設定をクリアする場合がある。この場合、継続して分岐命令実行時に割込みが上がるように、対象プログラムへの復帰時の実行コンテキストを再設定する必要がある。

【 0 0 4 2 】

(性能情報収集部)

性能情報収集部 1 7 は、命令分析部 1 6 から得た分岐元アドレス、分岐先アドレス及び分岐命令種別に加え、OS 8 上の実行体識別情報(実行体 ID)を取得する。即ち、性能情報収集部 1 7 は、割込発生時に、OS 8 によって提供されるシステムインターフェイスにアクセスすることによって、実行体 ID(プロセス ID、スレッド ID 又はタスク ID)を取得する。或いは、性能情報収集部 1 7 は、割込発生時に、OS 8 の制御情報に直接アクセスして実行体 ID を取得するようにしても良い。

【 0 0 4 3 】

続いて、性能情報収集部 1 7 は、取得した実行体 ID を用いてサブルーチンの実行体 1 9 を識別する。これによって、実行体 1 9 毎に性能情報を収集することが可能となる。

【 0 0 4 4 】

なお、この例では、性能情報収集部 1 7 は、アプリケーション 9 の実行時に生成される全ての実行体 1 9 について性能情報を収集する。これに代えて、少なくとも 1 つの特定の実行体 1 9 について性能情報を収集するようにしても良い。

【 0 0 4 5 】

性能情報収集部 1 7 は、実行体 1 9 を特定すると、特定した実行体 1 9 によって実行されるサブルーチンの性能情報を記憶する性能情報記憶部としての制御表 1 8 を作成又は更新する。

【 0 0 4 6 】

(制御表)

図 3 は、性能情報収集部 1 7 によって作成・更新される各制御表 1 8 の構成例を示す図である。制御表 1 8 は、MM 3 又は二次記憶装置 4 上に作成される。図 3 において、制御表 1 8 は、1 つの実行体管理表(execution managing table: E

MT) 2 1 と、少なくとも 1 つのサブルーチン管理表(subroutine managing table: SMT) 2 2 と、必要に応じて作成される少なくとも 1 つの呼出管理表(call managing table: CMT) 2 3 とからなる。

【 0 0 4 7 】

EMT 2 1 は、特定された実行体 1 9 によって実行されるサブルーチンについての SMT 2 2 及び CMT 2 3 を管理するためのテーブルである。EMT 2 1 は、“実行体 ID”と、“次 EMT ポインタ”と、“現 SMT ポインタ”と、“先頭 SMT ポインタ”とを保持する。

【 0 0 4 8 】

“実行体 ID”は、実行体 1 9 の識別情報である。“次 EMT ポインタ”は、EMT 2 1 に隣接する他の制御表 1 8 中の EMT 2 1 のアドレスを示す。“現 SMT ポインタ”は、現在の性能情報収集部 1 7 のアクセス対象に相当する SMT 2 2 (現在呼び出されているサブルーチンに対応する SMT 2 2) のアドレスを示す。“先頭 SMT ポインタ”は、複数の SMT 2 2 がある場合に、これらの先頭に位置する SMT 2 2 のアドレスを示す。

【 0 0 4 9 】

SMT 2 2 は、各サブルーチンの性能情報を管理するためのテーブルであり、実行体 1 9 によって実行されるサブルーチン毎に作成される。即ち、SMT 2 2 は、1 つの EMT 2 1 に対して、その EMT 2 1 によって管理される実行体 1 9 によって実行されるサブルーチンの数だけ作成される。

【 0 0 5 0 】

SMT 2 2 が複数作成された場合には、SMT 2 2 間は相互リンクで接続され、SMT 2 2 間を行ったり来たりすることができる。また、ある SMT 2 2 から他の SMT 2 2 を辿って目的の SMT 2 2 へ到達することができる。SMT 2 2 間の双方向リンクはサブルーチンの呼出しが発生した場合に相互に連結され、復帰時に解消される。

【 0 0 5 1 】

各 SMT 2 2 は、“サブルーチンアドレス”と、“SMT 双方向リンクポインタ”と、“先頭 CMT ポインタ”と、“現 CMT ポインタ”と、“呼出回数”と

、“累計実行時間”と、“最後の呼出時刻”とを保持する。“呼出回数”，“累計実行時間”及び“最後の呼出時刻”がサブルーチンの性能情報(プロフィール)に相当する。

【 0 0 5 2 】

“サブルーチンアドレス”は、EMT 2 1 の実行体 I D で特定される実行体 1 9 によって実行されるサブルーチンのアドレスを示す。“SMT 双方向リンクポインタ”は、複数の SMT 2 2 を相互リンクで接続した場合に設定されるポインタであり、SMT 2 2 自身のアドレスと、この SMT 2 2 に相互リンクが設定されている少なくとも 1 つの他の SMT 2 2 のアドレスとを示す。

【 0 0 5 3 】

“先頭 CMT ポインタ”は、この SMT 2 2 の下位に存する少なくとも 1 つの CMT 2 3 のうち、先頭に位置する CMT 2 3 のアドレスを示す。“現 CMT ポインタ”は、性能情報収集部 1 7 の現在のアクセス対象に相当する CMT 2 3 のアドレスを示す。

【 0 0 5 4 】

“呼出回数”は、EMT 2 1 の実行体 I D で特定される実行体 1 9 からの当該サブルーチンの呼出回数を示す。“累計実行時間”は、当該実行体 1 9 による当該サブルーチンの実行時間の合計値を示す。“最後の呼出時刻”は、当該実行体 1 9 によって当該サブルーチンが最後に呼び出された時刻である。

【 0 0 5 5 】

CMT 2 3 は、あるサブルーチンから他のサブルーチンが呼び出された(ネストした)場合に、呼出元のサブルーチンと呼出先のサブルーチンとの関係(呼出関係)と、呼出先のサブルーチンに対する性能情報を管理するために作成されるテーブルである。

【 0 0 5 6 】

CMT 2 3 は、1 つの SMT 2 2 で管理されるサブルーチンによって呼び出される他のサブルーチンの数を最低数として作成される。現在或るサブルーチンから他のサブルーチンが呼び出されている場合、対応する CMT 2 3 は、呼出元のサブルーチンに対応する SMT 2 2 中の“現 CMT ポインタ”で示される。

【 0 0 5 7 】

各 CMT 2 3 は、“分岐元アドレス”と、“分岐先アドレス”と、“呼出先 SMT ポインタ”と、“呼出回数”と、“累計実行時間”と、“最後の呼出時刻”と、“次 CMT ポインタ”とを保持する。“呼出回数”，“累計実行時間”及び“最後の呼出時刻”がサブルーチンの性能情報(プロフィール)に相当する。

【 0 0 5 8 】

“分岐元アドレス”は、呼出元に相当するサブルーチンのアドレスを示す。“分岐先アドレス”は、呼出先に相当するサブルーチンのアドレスを示す。“呼出先 SMT ポインタ”は、呼出先のサブルーチンに対応する SMT 2 2 のアドレスを示す。これらによって、サブルーチン間の呼出関係が保持・管理される。

【 0 0 5 9 】

“呼出回数”は、呼出元のサブルーチンから呼出先のサブルーチンが呼び出された回数を示す。“累計実行時間”は、呼出先のサブルーチンの実行時間の合計値を示す。“最後の呼出時刻”は、呼出元のサブルーチンから呼出先のサブルーチンを最後に呼び出した時刻を示す。“次の CMT ポインタ”は、この CMT 2 3 の次に配置された(隣接する)他の CMT 2 3 のアドレスを示す。

【 0 0 6 0 】

図 3 に示す例では、或る実行体 1 9 (例えば、図 2 に示す実行体 1 9 A)についての EMT 2 1 (EMT 2 1 A)が作成され、EMT 2 1 A で管理される実行体 1 9 A によって実行される各サブルーチンの性能情報を管理する SMT 2 2 として、各 SMT 2 2 A, 2 2 B, 2 2 C が作成されている。

【 0 0 6 1 】

さらに、SMT 2 2 A で管理されるサブルーチンに呼び出される他のサブルーチンの性能情報を管理するための CMT 2 3 A と、SMT 2 2 B で管理されるサブルーチンによって呼び出される他のサブルーチンの性能情報を管理するための CMT 2 3 B が作成されている。

【 0 0 6 2 】

EMT 2 1 A は、“次 EMT ポインタ”でアクセス可能な他の実行体 1 9 を管理する EMT 2 1 B に接続されており、性能情報収集部 1 7 は、EMT 2 1 A を

通じて次の EMT 2 1 B にアクセスすることができる。

【 0 0 6 3 】

SMT 2 2 A は、EMT 2 1 A に対する先頭 SMT に相当する。SMT 2 2 A は、SMT 2 A の次の SMT に相当する SMT 2 2 B に相互リンクで接続され、SMT 2 2 B は、SMT 2 2 B の次の SMT に相当する SMT 2 2 C に相互リンクで接続される。

【 0 0 6 4 】

性能情報収集部 1 7 は、所望の SMT 2 2 にアクセスする場合には、EMT 2 1 A の “先頭 SMT ポインタ” を用いて SMT 2 2 A にアクセスし、続いて、SMT 2 2 A, 2 2 B, 2 2 C の “SMT 双方向リンクポインタ” を用いて所望の SMT のアドレスを取得し、取得したアドレスを EMT 2 1 A の “現 SMT ポインタ” に設定する。

【 0 0 6 5 】

これによって、性能情報収集部 1 7 は、EMT 2 1 A から所望の SMT 2 2 へ直接アクセスすることができる。図 3 には、EMT 2 1 A の “現 SMT ポインタ” として、SMT 2 2 C のアドレスが保持されている場合の様子が示されている。

【 0 0 6 6 】

性能情報収集部 1 7 は、或る SMT 2 2 から所望の CMT 2 3 にアクセスする場合には、当該 SMT 2 2 の “先頭 CMT ポインタ” を用いて先頭 CMT に相当する CMT 2 3 にアクセスする。

【 0 0 6 7 】

続いて、性能情報収集部 1 7 は、先頭 CMT が所望の CMT 2 3 でない場合には、先頭 CMT 中の “次 CMT ポインタ” を用いて次の CMT 2 3 にアクセスする。このような処理を繰り返すことにより、所望の CMT 2 3 にアクセスすることができ、且つ SMT 2 2 の “現 CMT ポインタ” に所望の CMT 2 3 のアドレスを設定することができる。

【 0 0 6 8 】

図 3 には、SMT 2 2 A (SMT 2 2 B) の “先頭 CMT ポインタ” 及び “現 C

MTポインタ”としてCMT 2 3 A (CMT 2 3 B)のアドレスが保持されている場合の様子が示されている。

【 0 0 6 9 】

制御表 1 8 が上記データ構造を有することにより、各実行体 1 9 によって実行されるサブルーチンの性能情報が実行体 1 9 毎に保管される。また、各 SMT 2 2 によって、或る実行体によって実行されるサブルーチンの性能情報がサブルーチン毎に保管される。さらに、各 CMT 2 3 によって、各サブルーチンから呼び出され実行されるサブルーチンの性能情報と、サブルーチン間の呼出関係とが保管される。

【 0 0 7 0 】

なお、図 3 に示す例では、サブルーチンのネスティングが 1 回である場合の例を示しているが、ネスティングが 2 回以上行われる場合には、さらに下位の CMT 2 3 が作成される。例えば、CMT 2 3 A で管理される呼出先のサブルーチンから他のサブルーチンが呼び出される場合には、この呼出関係と呼び出された他のサブルーチンの性能情報を保管する下位の CMT 2 3 が作成される。

【 0 0 7 1 】

さらに、図 3 に例示した制御表 1 8 は、いわゆる「再帰(recursion)」を考慮しない場合(即ち、サブルーチン自身への再帰は単に分岐とみなし、呼出元に該当するメインルーチン又はサブルーチンに復帰する場合にのみカウントする)のデータ構造例を示す。これに代えて、制御表 1 8 を「再帰」を考慮したデータ構造に拡張することもできる。また、本発明における制御表 1 8 のデータ構造は、任意の構成を採用することができ、検索の効率化を可能とした二次的なハッシュ検索データの構築、制御表間の相互リンク構造の構築などを行なうことが可能である。

【 0 0 7 2 】

なお、制御表 1 8 は、サブルーチンを実行するタスク又はプロセス毎に作成されるようにしても良く、サブルーチンを実行するスレッド毎に作成されるようにしても良い。

【 0 0 7 3 】

〈動作例〉

次に、上述したプログラム性能情報収集装置の動作例を説明する。図 2 において、最初に、実行環境設定部 1 5 が、CPU 2 に対し、上述した設定を施す(図 2 (A)参照)。その後、CPU 2 が、OS 8 上でアプリケーション 9 を実行する。これによって、図 2 (B)に示すように、アプリケーション 9 が OS 8 上で実行され、アプリケーション 9 中のサブルーチンが実行される場合には、サブルーチンの実行体 1 9 が生成され、実行体 1 9 がサブルーチンを実行する。

【 0 0 7 4 】

アプリケーション 9 の実行時において、分岐命令が実行されると、CPU 2 が割込を発生させるとともに、制御を命令分析部 1 6 に渡す。これによって、命令分析部 1 6 が割込処理を開始する。図 4 は、命令分析部 1 6 の処理を示すフローチャートである。

【 0 0 7 5 】

図 4 において、最初に、命令分析部 1 6 は、実行された分岐命令の分岐元アドレスと分岐先アドレスとを CPU 2 から受け取る(ステップ S 1)。次に、命令分析部 1 6 は、分岐元アドレスの命令コードを MM 3 から読み出し(ステップ S 2)、命令コードの命令フォーマットをデコードすることによって分岐命令の種別を判別する(ステップ S 3)。

【 0 0 7 6 】

続いて、命令分析部 1 6 は、分岐命令を判別した結果、命令種別がサブルーチンの呼出命令(CALL 命令)又は復帰命令(RETURN 命令)であるか否かを判定する(ステップ S 4)。即ち、命令分析部 1 6 は、分岐命令がサブルーチンの実行に関する命令(呼出命令又は復帰命令)であるか否かを判別する。

【 0 0 7 7 】

このとき、分岐命令が呼出命令又は復帰命令である場合には、分岐元アドレス、分岐先アドレス及び命令種別の判別結果を性能情報収集部 1 7 に与える(ステップ S 5)。これに対し、命令分析部 1 6 は、命令種別が呼出命令及び復帰命令以外である場合には、割込処理を終了し、制御を CPU 2 に戻す(ステップ S 6)。

【 0 0 7 8 】

処理がステップ S 5 へ進んだ場合には、命令分析部 1 6 に続いて、性能情報収集部 1 7 による割込処理が実行される。図 5 , 6 , 7 は、性能情報収集部 1 7 の処理を示すフローチャートである。

【 0 0 7 9 】

図 5 において、性能情報収集部 1 7 は、命令分析部 1 6 から分岐元アドレス、分岐先アドレス及び命令種別の判別結果を受け取ると、サブルーチンの実行体 I D を O S 8 から取得し、サブルーチンの実行体 1 9 を特定する(ステップ S 1 0 1)。

【 0 0 8 0 】

次に、性能情報収集部 1 7 は、命令種別の判別結果がサブルーチンの呼出命令か否かを判定する(ステップ S 1 0 2)。このとき、判別結果が呼出命令である場合には、処理がステップ S 1 0 3 に進み、判別結果が呼出命令でない場合(復帰命令である場合)には、処理が図 7 のステップ S 1 2 5 へ進む。

【 0 0 8 1 】

ステップ S 1 0 3 では、性能情報収集部 1 7 は、呼出命令がサブルーチン中からの呼出命令、即ち或るサブルーチンから他のサブルーチンを呼び出す呼出命令か否かを判定する。具体的には、性能情報収集部 1 7 は、命令分析部 1 6 から受け取った分岐元アドレスがサブルーチンのアドレスを示すか否かを判定する。

【 0 0 8 2 】

このとき、分岐元アドレスがサブルーチンのアドレスでない場合(メインルーチンのアドレスである場合)には、性能情報収集部 1 7 は、呼出命令がサブルーチン中からの呼出命令ではないものとして、処理をステップ S 1 0 4 に進める。これに対し、分岐元アドレスがサブルーチンのアドレスである場合には、性能情報収集部 1 7 は、呼出命令がサブルーチン中からの呼出命令であるものとして、処理を図 6 に示すステップ S 1 1 5 に進める。

【 0 0 8 3 】

ステップ 1 0 4 に処理が進んだ場合には、性能情報収集部 1 7 は、ステップ S 1 0 1 にて取得した実行体 I D を保持した E M T 2 1 を含む制御表 1 8 を検索す

る。即ち、性能情報収集部 1 7 は、ステップ S 1 0 1 にて特定した実行体 1 9 に対応する EMT 2 1 があるか否かを判定する。このとき、対応する EMT 2 1 がない場合には、性能情報収集部 1 7 は、処理をステップ S 1 0 5 に進め、対応する EMT 2 1 がある場合には、処理をステップ S 1 0 7 に進める。

【 0 0 8 4 】

ステップ S 1 0 5 では、性能情報収集部 1 7 は、実行体 1 9 に対応する制御表 1 8 がないものとして、実行体 1 9 に対応する EMT 2 1 を新規に作成する。これにより、実行体 1 9 に対応した新規の制御表 1 8 が作成される。続いて、性能情報収集部 1 7 は、作成した EMT 2 1 にステップ S 1 0 1 にて取得した実行体 ID を設定(登録)する(ステップ S 1 0 6)。その後、処理がステップ S 1 0 7 に進む。

【 0 0 8 5 】

ステップ S 1 0 7 では、性能情報収集部 1 7 は、呼出命令によって呼び出されたサブルーチンに対応する SMT 2 2 があるか否かを判定する。即ち、性能情報収集部 1 7 は、命令分析部 1 6 から受け取った分岐先アドレスを“サブルーチンアドレス”として保持した SMT 2 2 を検索する。このとき、該当する SMT 2 2 がある場合には、処理がステップ S 1 1 3 に進み、該当する SMT 2 2 がない場合には、処理がステップ S 1 0 8 に進む。

【 0 0 8 6 】

ステップ S 1 0 8 では、性能情報収集部 1 7 は、サブルーチンに対応する新規な SMT 2 2 を作成する。続いて、性能情報収集部 1 7 は、分岐先アドレスを“サブルーチンアドレス”として、作成した SMT 2 2 に設定(登録)する(ステップ S 1 0 9)。

【 0 0 8 7 】

次に、性能情報収集部 1 7 は、新規に作成した SMT 2 2 が先頭 SMT に相当するか否かを判定する(ステップ S 1 1 0)。このとき、SMT 2 2 が先頭 SMT に相当する場合には、処理がステップ S 1 1 1 に進み、そうでない場合には、処理がステップ S 1 1 2 に進む。

【 0 0 8 8 】

ステップ S 1 1 1 に処理が進んだ場合には、性能情報収集部 1 7 は、新規に作成した SMT 2 2 のアドレスを、“先頭 SMT ポインタ”として、対応する EMT 2 1 に設定する。その後、処理がステップ S 1 1 3 に進む。

【 0 0 8 9 】

ステップ S 1 1 2 に処理が進んだ場合には、性能情報収集部 1 7 は、既存の SMT 2 2 の何れかと新規に作成した SMT 2 2 とのリンクを設定し、各 SMT 2 2 の“SMT 双方向リンクポインタ”に適宜のアドレスを設定(登録)する。その後、処理がステップ S 1 1 3 に進む。

【 0 0 9 0 】

ステップ S 1 1 3 では、性能情報収集部 1 7 は、呼び出されたサブルーチンのアドレスが登録された SMT 2 2 のアドレスを、“現 SMT ポインタ”として、対応する EMT 2 1 に設定(登録)する。

【 0 0 9 1 】

続いて、性能情報収集部 1 7 は、例えば、現在の時刻を、“最後の呼出時刻”として、該当する SMT 2 2 に格納する(ステップ S 1 1 4)。なお、時刻は、コンピュータ 1 に搭載された図示せぬ内蔵時計から受け取ることができる。その後、性能情報収集部 1 7 は、割込処理を終了し、制御を CPU 2 に戻す。その後、実行体 1 9 によってサブルーチンが実行される。

【 0 0 9 2 】

ところで、処理が図 6 のステップ S 1 1 5 に進んだ場合には、性能情報収集部 1 7 は、呼出先のサブルーチンに対応する CMT 2 3 があるか否かを判定する。即ち、性能情報収集部 1 7 は、命令分析部 1 6 から受け取った分岐先アドレスを“分岐先アドレス”として保持した CMT 2 3 を検索する。

【 0 0 9 3 】

このとき、性能情報収集部 1 7 は、該当する CMT 2 3 が見つかった場合には、処理をステップ S 1 1 8 に進め、該当する CMT 2 3 が見つからない場合には、呼出先のサブルーチンに対応する CMT 2 3 がないものとして、処理をステップ S 1 1 6 に進める。

【 0 0 9 4 】

ステップ S 1 1 6 では、性能情報収集部 1 7 は、呼出先のサブルーチンに対応する CMT 2 3 を新規に作成する。続いて、性能情報収集部 1 7 は、命令分析部 1 6 から受け取った分岐元アドレスを、“呼出元のサブルーチンアドレス”として、作成した CMT 2 3 に設定(登録)し、且つ命令分析部 1 6 から受け取った分岐先アドレスを、“呼出先のサブルーチンアドレス”として、作成した CMT 2 3 に設定(登録)する(ステップ S 1 1 7)。これによって、サブルーチンのネスティングが行われた場合におけるサブルーチン間の呼出関係が CMT 2 3 に格納される。その後、処理がステップ S 1 1 8 に進む。

【 0 0 9 5 】

ステップ S 1 1 8 では、性能情報収集部 1 7 は、作成した CMT 2 3 が“先頭 CMT”に該当するか否かを判定する。このとき、当該 CMT 2 3 が“先頭 CMT”に該当する場合には、性能情報収集部 1 7 は、当該 CMT 2 3 のアドレスを、“先頭 CMT ポインタ”として、呼出元のサブルーチンに対応する SMT 2 2 に設定(登録)し(ステップ S 1 1 9)、その後、処理をステップ S 1 2 1 へ進める。

【 0 0 9 6 】

これに対し、当該 CMT 2 3 が“先頭 CMT”に該当しない場合には、性能情報収集部 1 7 は、当該 CMT 2 3 のアドレスを、“次の CMT ポインタ”として、当該 CMT 2 3 の 1 つ前に配置された他の CMT 2 3 に設定(登録)し(ステップ S 1 2 0)、その後、処理をステップ S 1 2 1 へ進める。

【 0 0 9 7 】

ステップ S 1 2 1 では、性能情報収集部 1 7 は、呼出先のサブルーチンに対応する SMT 2 2 があるか否かを判定する。即ち、性能情報収集部 1 7 は、当該 CMT 2 3 に“分岐先アドレス(呼出先のサブルーチンアドレス)”として設定したアドレスが“サブルーチンアドレス”として登録された SMT 2 2 を、当該制御表 1 8 から検索する。

【 0 0 9 8 】

このとき、該当する SMT 2 2 がない場合には、性能情報収集部 1 7 は、処理をステップ S 1 2 3 に進める。これに対し、該当する SMT 2 2 が見つかった場

合には、性能情報収集部 1 7 は、見つかった SMT 2 2 のアドレスを、“呼出先 SMT ポインタ”として、当該 CMT 2 3 に設定(登録)し(ステップ S 1 1 9)、その後、処理をステップ S 1 2 3 に進める。

【 0 0 9 9 】

ステップ S 1 2 3 では、性能情報収集部 1 7 は、当該 CMT 2 3 のアドレスを、“現 CMT ポインタ”として、呼出元のサブルーチンに対応する SMT 2 2 に設定(登録)する。

【 0 1 0 0 】

続いて、性能情報収集部 1 7 は、例えば、現時刻を取得し、取得した現時刻を、“最後の呼出時刻”として、当該 CMT 2 3 に格納する(ステップ S 1 2 4)。その後、性能情報収集部 1 7 は、割込処理を終了し、制御を CPU 2 に戻す。その後、ネスティングによって呼び出されたサブルーチンが実行体 1 9 によって実行される。

【 0 1 0 1 】

ところで、分岐命令の判別結果が復帰命令であり、処理が図 7 に示したステップ S 1 2 5 に進んだ場合には、性能情報収集部 1 7 は、復帰命令が他のサブルーチンへの復帰命令か否かを判定する。具体的には、性能情報収集部 1 7 は、命令分析部 1 6 から受け取った分岐先アドレスがサブルーチンのアドレスを示すか否かを判定する。

【 0 1 0 2 】

このとき、分岐先アドレスがサブルーチンのアドレスでない場合(メインルーチンのアドレスである場合)には、性能情報収集部 1 7 は、復帰命令が他のサブルーチンへの復帰命令ではないものとして、処理をステップ S 1 2 6 に進める。これに対し、分岐先アドレスがサブルーチンのアドレスである場合には、性能情報収集部 1 7 は、復帰命令が他のサブルーチンへの復帰命令であるものとして、処理をステップ S 1 3 0 に進める。

【 0 1 0 3 】

ステップ S 1 2 6 では、性能情報収集部 1 7 は、該当する SMT 2 2 の“呼出回数”を 1 インクリメントする。続いて、性能情報収集部 1 7 は、現在の時刻を

取得し(ステップ S 1 2 7)、現時刻と、当該 SMT 2 2 に格納された“最後の呼出時刻”とから、サブルーチンの実行時間を算出する(ステップ S 1 2 8)。

【 0 1 0 4 】

続いて、性能情報収集部 1 7 は、当該 SMT 2 2 の“累計実行時間”の値に、ステップ S 1 2 8 にて算出した実行時間を加算することによって、“累計実行時間”を更新する(ステップ S 1 2 9)。その後、性能情報収集部 1 7 は、割込処理を終了し、制御を CPU 2 に戻す。

【 0 1 0 5 】

一方、ステップ S 1 3 0 に処理が進んだ場合には、性能情報収集部 1 7 は、該当する CMT 2 3 の“呼出回数”を 1 インクリメントする。続いて、性能情報収集部 1 7 は、現在の時刻を取得し(ステップ S 1 3 1)、現時刻と、当該 CMT 2 3 に格納された“最後の呼出時刻”とから、サブルーチンの実行時間を算出する(ステップ S 1 3 2)。

【 0 1 0 6 】

続いて、性能情報収集部 1 7 は、当該 CMT 2 3 の“累計実行時間”の値に、ステップ S 1 3 2 にて算出した実行時間を加算することによって、“累計実行時間”を更新する(ステップ S 1 3 3)。その後、性能情報収集部 1 7 は、割込処理を終了し、制御を CPU 2 に戻す。

【 0 1 0 7 】

＜実施形態の作用＞

上述した実施形態によるプログラム性能情報収集装置によると、実行環境設定部 1 5 による CPU 2 の設定により、アプリケーション 9 の実行時に分岐命令が実行されると、割込が発生し、制御が CPU 2 から命令分析部 1 6 へ渡される。

【 0 1 0 8 】

その後、命令分析部 1 6 及び性能情報収集部 1 7 によって、サブルーチンの実行体 1 9 毎に制御表 1 8 が作成され、各制御表 1 8 において、実行体 1 9 によって実行される各サブルーチンの性能情報(プロフィール)が保管される。

【 0 1 0 9 】

即ち、制御表 1 8 における各 SMT 2 2 において、メインルーチンから呼び出

されたサブルーチンの呼出回数、累計実行時間及び最後の呼出時刻が性能情報として保持される。また、制御表 1 8 における各 CMT 2 3 において、或るサブルーチンから呼び出された他のサブルーチンの呼出回数、累計実行時間及び最後の呼出時刻が性能情報として保持される。また、各 CMT 2 3 には、呼出元のサブルーチンと呼出先のサブルーチンとの夫々のアドレスが格納されることにより、サブルーチン間の呼出関係を示す情報(呼出関係情報)が保持される。

【 0 1 1 0 】

上記した命令分析部 1 6 及び性能情報収集部 1 7 の処理(分析プログラム 1 0 の実行による処理)は、CPU 2 の機能を用いた割込により行うため、アプリケーション 9 に対して何ら改変を加える必要がない。従って、制御コードをプログラムに埋め込むことによるオーバヘッドの増加や、プログラムの改変を起因とするプログラムの不動作といった従来の問題は生じない。

【 0 1 1 1 】

アプリケーション 9 の実行が終了すると、MM 3 又は二次記憶装置 4 上には、アプリケーション 9 に含まれた複数のサブルーチンについての性能情報及び呼出関係情報を保持した複数の制御表 1 8 が作成される。コンピュータ 1 のオペレータは、入力装置 1 1 の操作によって、各制御表 1 8 の保持内容をディスプレイ 7 に表示したり、図示せぬプリンタによって紙等のシートに印刷したりすることができる。そして、ディスプレイ 7 に表示されたり、シートに印刷されたりした制御表 1 8 の保持内容は、アプリケーション 9 の改良や、新規のアプリケーションプログラムの作成の資料として使用される。

【 0 1 1 2 】

なお、本実施形態では、或るサブルーチンについて、メインルーチンから呼び出された場合の性能情報を SMT 2 2 に格納し、他のサブルーチンから呼び出された場合の性能情報を CMT 2 3 に格納した。これらの SMT 2 2 及び CMT 2 3 中の呼出回数同士、累計実行時間同士を夫々加算することにより、或るサブルーチンのアプリケーション 9 の実行時における累計実行時間を得ることができる。

【 0 1 1 3 】

これに対し、SMT 2 2 には、或るサブルーチンの呼出回数及び累計実行時間呼出回数を格納し、CMT 2 3 に SMT 2 2 に格納された呼出回数及び累計実行時間のうち、他のサブルーチンから呼び出された場合の呼出回数及び累計実行時間を格納するようにしても良い。この場合には、SMT 2 2 に格納された呼出回数及び累計実行時間から CMT 2 3 に格納された呼出回数及び累計実行時間を夫々減算することにより、メインルーチンから呼び出された場合の呼出回数及び累計実行時間を得ることができる。

【 0 1 1 4 】

〔第 2 実施形態〕

次に、本発明の第 2 実施形態によるプログラム性能情報収集装置について説明する。第 2 実施形態は、第 1 実施形態の構成に加え、サブルーチンの性能情報として、オーバーヘッドを制御表 1 8 に格納するようにしたものである。

【 0 1 1 5 】

即ち、第 2 実施形態では、性能情報収集部 1 7 は、アプリケーション 9 の実行時において、分岐命令の実行を契機とする割込処理の発生回数を、各サブルーチンについて、その呼出元アドレス(分岐元アドレス)に従い、各 SMT 2 3 及び各 CMT 2 3 に夫々格納する。

【 0 1 1 6 】

即ち、性能情報収集部は、各サブルーチンに対する呼出命令、復帰命令及びジャンプ命令の発生回数の合計値を、そのサブルーチンがメインルーチンから呼び出された場合と、他のサブルーチンとから呼び出された場合とに分けて、該当する SMT 2 2 及び CMT 2 3 に格納する。また、性能情報収集部 1 7 は、平均オーバーヘッド時間を算出する。図 8 に第 2 実施形態における制御表 1 8 A の構成例を示す。

【 0 1 1 7 】

その後、アプリケーション 9 の実行中、或いは実行終了後に、各 SMT 2 2 及び各 CMT 2 3 に格納された割込処理の発生回数に、算出した平均オーバーヘッド時間を乗じ、この結果を各サブルーチンの実行時におけるオーバーヘッド時間として各 SMT 2 2 及び各 CMT 2 3 に格納する。

【 0 1 1 8 】

第 2 実施形態によれば、サブルーチンの性能情報として、さらにオーバヘッドが各制御表 1 8 に格納されるので、第 1 実施形態よりも詳細なサブルーチンの性能情報(プロフィール)を得ることができ、性能情報の精度を向上させることができる。

【 0 1 1 9 】

〔付記〕

本発明は、以下のように特定することができる。

(付記 1) プログラムに含まれたサブルーチンの性能情報を収集する装置であって、前記プログラムの実行中に分岐命令が実行されたときに発生する割込によって、前記サブルーチンの性能情報を収集する、プログラム性能情報収集装置。

(付記 2) サブルーチンの実行体毎に前記性能情報を収集する、付記 1 記載のプログラム性能情報収集装置。

(付記 3) 特定の実行体が複数のサブルーチンを実行する場合に、前記各サブルーチンの性能情報を個別に収集する、付記 2 記載のプログラム性能情報収集装置。

(付記 4) 或るサブルーチンについて、メインルーチンから呼び出された場合の性能情報と、他のサブルーチンから呼び出された場合の性能情報とを個別に収集する、付記 3 記載のプログラム性能情報収集装置。

(付記 5) 前記他のサブルーチンから呼び出された場合の性能情報を、他のサブルーチンと呼び出されたサブルーチンとの関係を示す呼出関係情報と関連づけて保持する、付記 4 記載のプログラム性能情報収集装置。

(付記 6) 前記性能情報は、サブルーチンの累計実行時間、呼出回数、最後に呼び出された時刻、オーバヘッドのうちの少なくとも 1 つである、付記 1 記載のプログラム性能情報収集装置。

(付記 7) 前記サブルーチンの性能情報を保持する性能情報記憶部と、前記割込が発生した場合に、実行された分岐命令が前記サブルーチンの実行に関する命令か否かを判別する命令分析部と、前記命令分析部によって前記分岐命令が前記サブルーチンの実行に関する命令と判別した場合に、前記サブルーチンの性能情報

を取得して前記性能情報記憶部に格納する性能情報収集部と、を備えた付記 1 記載のプログラム性能情報収集装置。

(付記 8) 前記性能情報記憶部は、前記サブルーチンの実行体毎に設けられ、前記性能情報収集部は、前記サブルーチンの実行体を特定し、特定した実行体に対応する性能情報記憶部に前記性能情報を格納する、付記 7 記載のプログラム性能情報収集装置。

(付記 9) 前記性能情報収集部は、特定の実行体によって複数のサブルーチンが実行される場合に、各サブルーチンの性能情報を、前記特定の実行体に対応する性能情報記憶部に個別に格納する、付記 8 記載のプログラム性能情報収集装置。

(付記 10) 前記性能情報収集部は、或るサブルーチンについて、メインルーチンから呼び出された場合の性能情報と、他のサブルーチンから呼び出された場合の性能情報とを、前記性能情報記憶部に個別に格納する、付記 9 記載のプログラム性能情報収集装置。

(付記 11) 前記性能情報収集部は、他のサブルーチンから呼び出された場合のサブルーチンの性能情報を、他のサブルーチンと呼び出されたサブルーチンとの関係を示す呼出関係情報と関連づけて前記性能情報記憶部に格納する、付記 10 記載のプログラム性能情報収集装置。

(付記 12) 前記性能情報を記憶する性能情報記憶部と、前記割込が発生した場合に、前記割込の発生源から分岐元アドレス及び分岐先アドレスとを受け取り、分岐元アドレスから命令コードを取得しデコードすることによって分岐命令の種別を判別する命令分析部と、判別された種別がサブルーチンの呼出命令又は復帰命令である場合に、分岐元アドレス、分岐先アドレス及び判別結果を前記命令分析部から受け取り、受け取った判別結果が呼出命令である場合には、受け取った分岐先アドレスを呼出命令に対応するサブルーチンアドレスとして前記性能情報記憶部に記憶するとともに、呼出命令に対応するサブルーチンの呼出時刻を前記分岐先アドレスに関連づけて前記性能情報記憶部に格納し、受け取った判別結果が復帰命令である場合には、復帰命令に対応するサブルーチンの復帰時刻を取得し、取得した復帰時刻と前記呼出時刻とから前記サブルーチンの実行時間を算出し、実行時間の累計値を前記性能情報として前記分岐先アドレスと関連づけて前

記性能情報記憶部に格納する、付記 1 記載のプログラム性能情報収集装置。

（付記 1 3）前記性能情報収集部は、サブルーチンの呼出回数を前記性能情報として前記分岐先アドレスと関連づけて前記性能情報記憶部に格納する、付記 1 2 記載のプログラム性能情報収集装置。

（付記 1 4）前記性能情報収集部は、前記性能情報として、サブルーチンのオーバーヘッドを取得し前記性能情報記憶部に格納する、付記 1 2 記載のプログラム性能情報収集装置。

（付記 1 5）前記性能情報収集部は、受け取った判別結果がサブルーチンの呼出命令である場合には、この呼出命令に対応するサブルーチンの実行体の識別情報を取得し、前記分岐先アドレスと関連づけて前記性能情報記憶部に格納する、付記 1 3 記載のプログラム性能情報収集装置。

（付記 1 6）前記性能情報収集部は、受け取った判別結果がサブルーチンの呼出命令であり、且つ受け取った分岐元アドレス及び分岐先アドレスがサブルーチンのアドレスを示す場合には、これらの分岐元アドレス及び分岐先アドレスを呼出元のサブルーチンと呼出先のサブルーチンとを示す呼出関係情報として前記性能情報記憶部に格納するとともに、呼出元のサブルーチンにおける呼出先のサブルーチンの累計実行時間と呼出回数との少なくとも一方を、前記性能情報として、前記呼出関係情報と関連づけて前記性能情報記憶部に格納する、付記 1 3 記載のプログラム性能情報収集装置。

（付記 1 7）前記プログラムの実行中に分岐命令が実行されたときに割込を発生させるように割込発生源の実行環境を設定する実行環境設定部をさらに備えた、付記 7 記載のプログラム性能情報収集装置。

（付記 1 8）分析対象のプログラムに含まれたサブルーチンの性能情報を収集する処理をコンピュータに実行させるプログラムを記録した記録媒体であって、コンピュータに、前記プログラムの実行中に分岐命令が実行されたことを契機として割込が発生した場合に、実行された分岐命令がサブルーチンの実行に関する命令か否かを判別するステップと、前記分岐命令がサブルーチンの実行に関する命令と判別された場合に、前記サブルーチンの性能情報を取得して性能情報記憶部に格納するステップと、を実行させるプログラムを記録したコンピュータ読取可

能な記録媒体。

（付記 1 9）前記サブルーチンの実行体を特定するステップと、実行体毎に設けられた複数の性能情報記憶部のうち、特定された実行体に対応する性能情報記憶部に前記性能情報を格納するステップと、をさらに実行させる前記プログラムを記録した付記 1 8 記載のコンピュータ読取可能な記録媒体。

（付記 2 0）特定の実行体によって複数のサブルーチンが実行される場合に、各サブルーチンの性能情報を、前記特定の実行体に対応する性能情報記憶部に個別に格納するステップ、をさらに実行させる前記プログラムを記録した付記 1 9 記載のコンピュータ読取可能な記録媒体。

（付記 2 1）前記各サブルーチンについて、メインルーチンから呼び出された場合の性能情報と、他のサブルーチンから呼び出された場合の性能情報とを、前記性能情報記憶部に個別に格納するステップ、をさらに実行させる前記プログラムを記録した付記 2 0 記載のコンピュータ読取可能な記録媒体。

（付記 2 2）他のサブルーチンから呼び出された場合のサブルーチンの性能情報を、他のサブルーチンと呼び出されたサブルーチンとの関係を示す呼出関係情報と関連づけて前記性能情報記憶部に格納するステップ、をさらに実行させる前記プログラムを記録した付記 2 1 記載のコンピュータ読取可能な記録媒体。

（付記 2 3）割込が発生した場合に前記割込の発生源から分岐元アドレス及び分岐先アドレスとを受け取るステップと、分岐元アドレスから命令コードを取得しデコードすることによって分岐命令の種別を判別するステップと、判別された種別がサブルーチンの呼出命令である場合に、受け取った分岐先アドレスを呼出命令に対応するサブルーチンアドレスとして前記性能情報記憶部に記憶するとともに、呼出命令に対応するサブルーチンの呼出時刻を前記分岐先アドレスに関連づけて前記性能情報記憶部に格納するステップと、判別された種別が復帰命令である場合に、復帰命令に対応するサブルーチンの復帰時刻を取得し、取得した復帰時刻と前記呼出時刻とから前記サブルーチンの実行時間を算出し、実行時間の累計値を前記性能情報として前記分岐先アドレスと関連づけて前記性能情報記憶部に格納するステップと、をさらに実行させる前記プログラムを記録した付記 1 8 記載のコンピュータ読取可能な記録媒体。

（付記 2 4）サブルーチンの呼出回数を前記性能情報として前記分岐先アドレスと関連づけて前記性能情報記憶部に格納するステップ、をさらに実行させる前記プログラムを記録した付記 2 3 記載のコンピュータ読取可能な記録媒体。

（付記 2 5）サブルーチンのオーバーヘッドを前記性能情報として前記分岐先アドレスと関連づけて前記性能情報記憶部に格納するステップ、をさらに実行させる前記プログラムを記録した付記 2 3 又は 2 4 記載のコンピュータ読取可能な記録媒体。

（付記 2 6）判別された種別がサブルーチンの呼出命令である場合に、呼出命令に対応するサブルーチンの実行体の識別情報を取得し、前記分岐先アドレスと関連づけて前記性能情報記憶部に格納するステップ、をさらに実行させる前記プログラムを記録した付記 2 3 記載のコンピュータ読取可能な記録媒体。

（付記 2 7）判別された種別がサブルーチンの呼出命令であり、且つ分岐元アドレス及び分岐先アドレスがサブルーチンのアドレスを示す場合に、これらの分岐元アドレス及び分岐先アドレスを呼出元のサブルーチンと呼出先のサブルーチンとを示す呼出関係情報として前記性能情報記憶部に格納するステップと、前記呼出元のサブルーチンにおける前記呼出先のサブルーチンの累計実行時間と呼出回数との少なくとも一方を、前記呼出関係情報と関連づけて前記性能情報記憶部に格納するステップと、をさらに実行させる付記 2 3 記載のコンピュータ読取可能な記録媒体。

（付記 2 8）前記プログラムの実行中に分岐命令が実行されたときに割込を発生させるように割込発生源の実行環境を設定するステップ、をさらに実行させる前記プログラムを記録した付記 1 8 記載のコンピュータ読取可能な記録媒体。

【 0 1 2 0 】

【発明の効果】

本発明によれば、プログラムに改変を加えることなくサブルーチンの性能情報を収集することができる。また、詳細なサブルーチンの性能情報を収集することができる。

【図面の簡単な説明】

【図 1】プログラム性能情報収集装置として機能するコンピュータのハードウェア

ア構成図

【図 2】 プログラム性能情報収集装置の機能ブロック図

【図 3】 制御表の構成例を示す図

【図 4】 命令分析部の処理を示すフローチャート

【図 5】 性能情報収集部の処理を示すフローチャート

【図 6】 性能情報収集部の処理を示すフローチャート

【図 7】 性能情報収集部の処理を示すフローチャート

【図 8】 第 2 実施形態における制御表の構成例を示す図

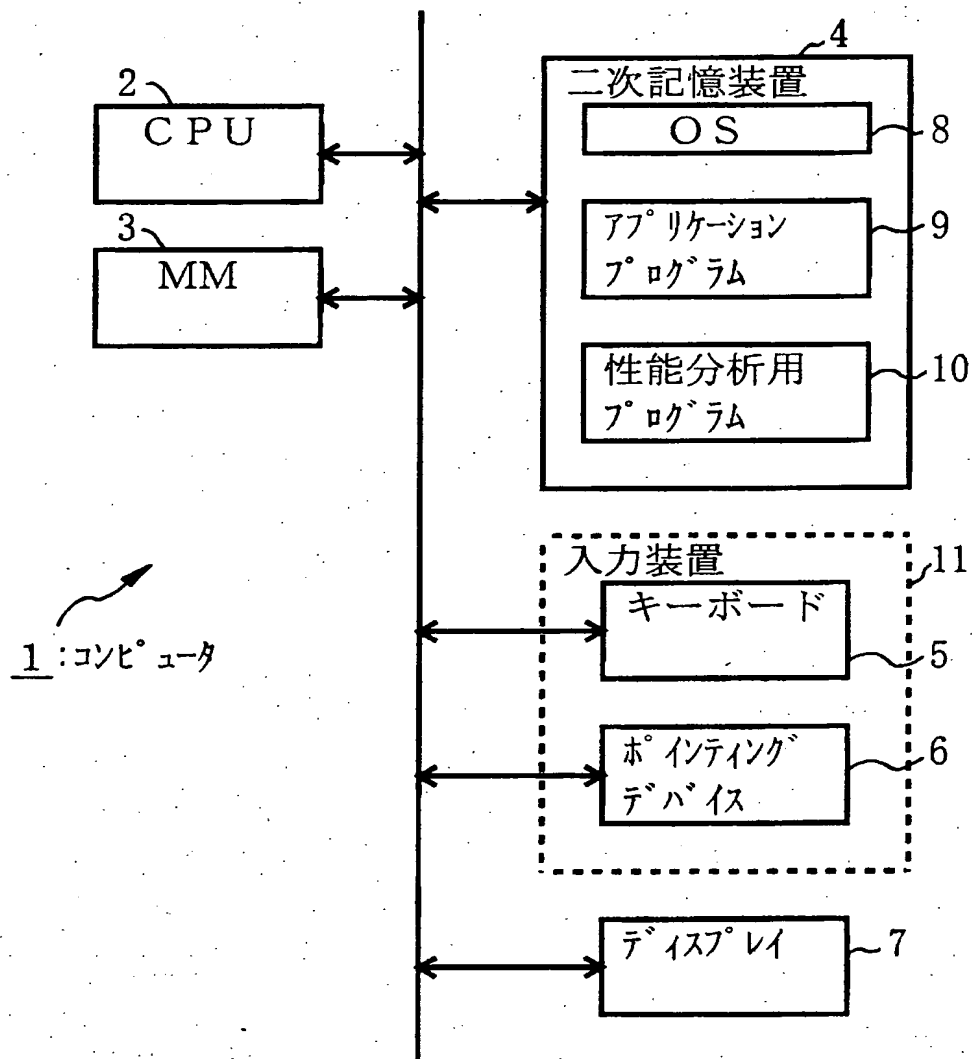
【符号の説明】

- 1 コンピュータ
- 2 CPU
- 3 メインメモリ
- 4 二次記憶装置
- 5 キーボード
- 6 ポインティングデバイス
- 7 ディスプレイ
- 8 オペレーティングシステム
- 9 アプリケーションプログラム
- 10 性能分析用プログラム
- 11 入力装置
- 15 実行環境設定部
- 16 命令分析部
- 17 性能情報収集部
- 18 制御表
- 21 実行体管理表(EMT)
- 22 サブルーチン管理表(SMT)
- 23 呼出管理表(CMT)

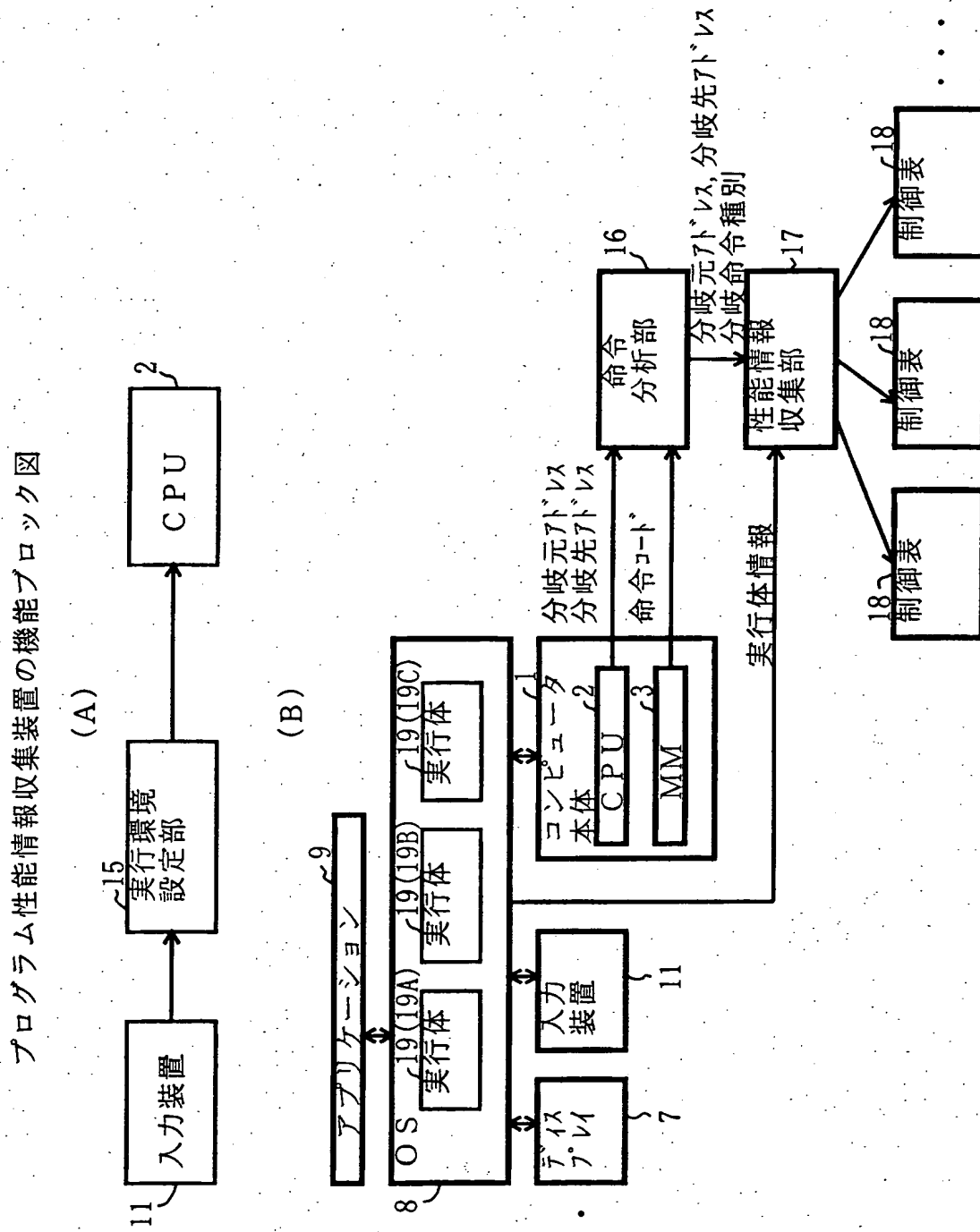
【書類名】 図面

【図 1】

プログラム性能情報収集装置として機能する
コンピュータのハードウェア構成図

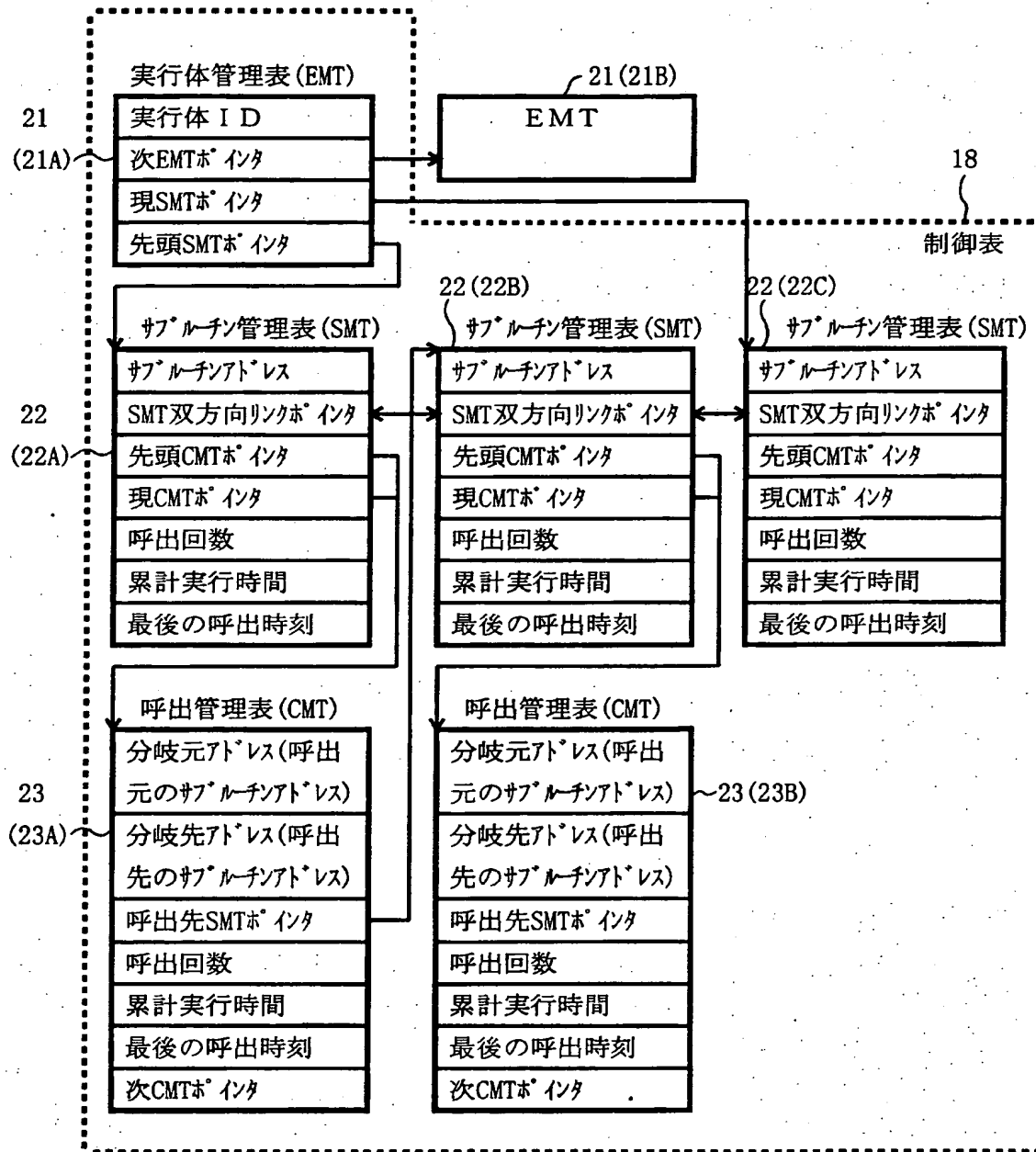


【图 2】



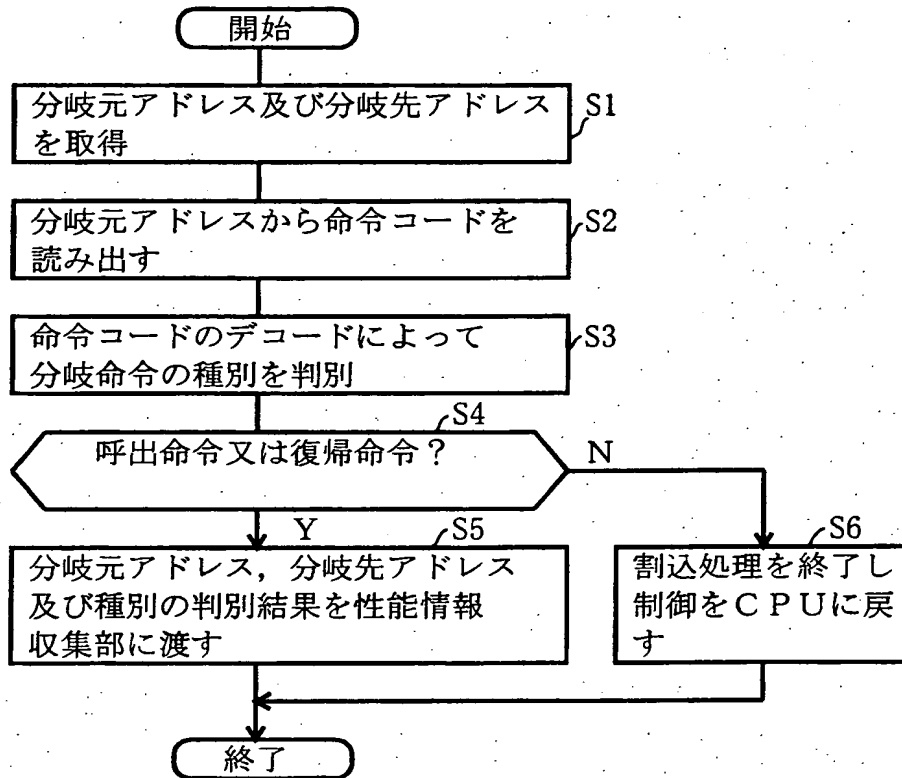
【図 3】

制御表の構成例を示す図



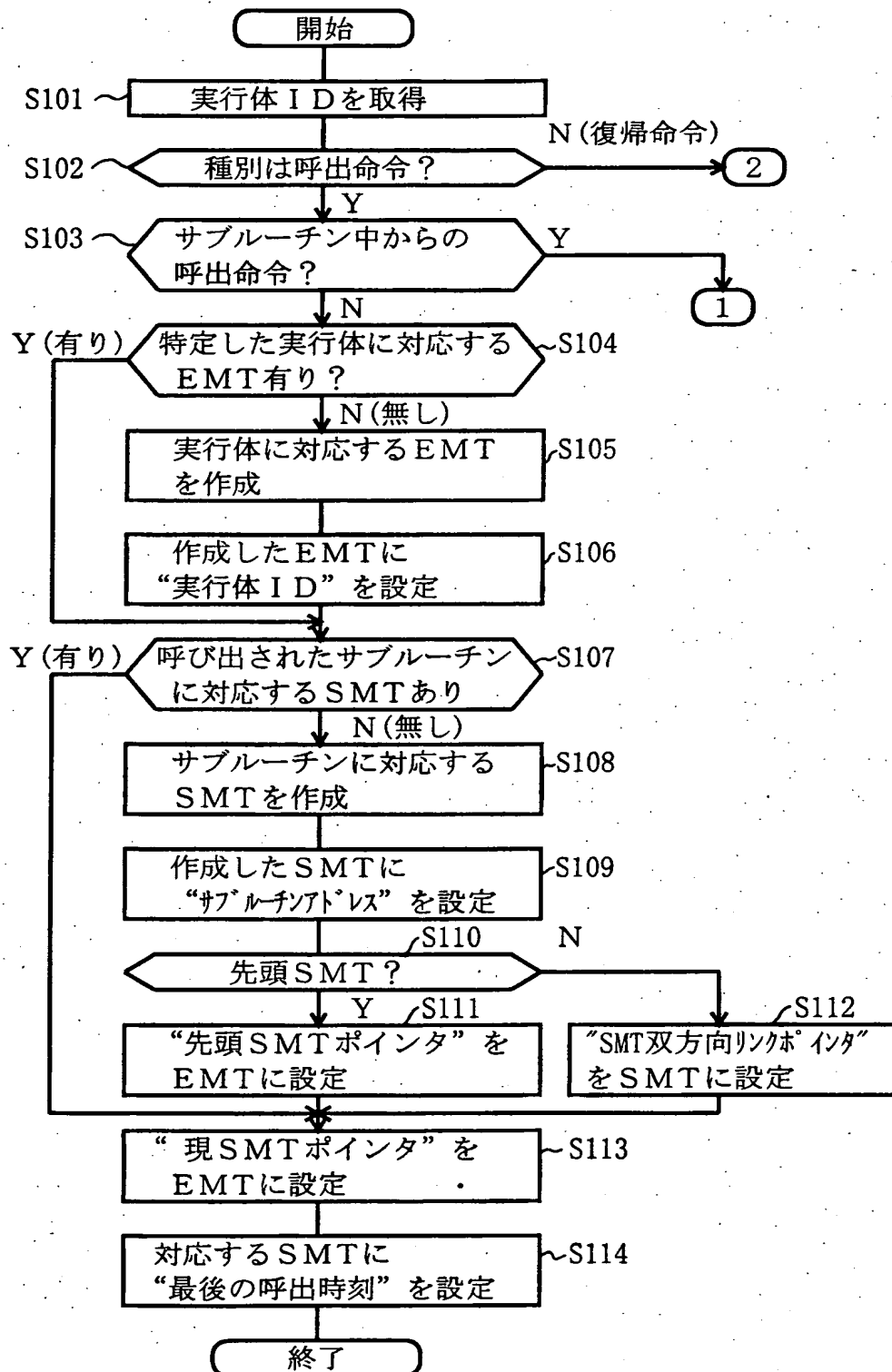
【図 4】

命令分析部の処理を示すフローチャート



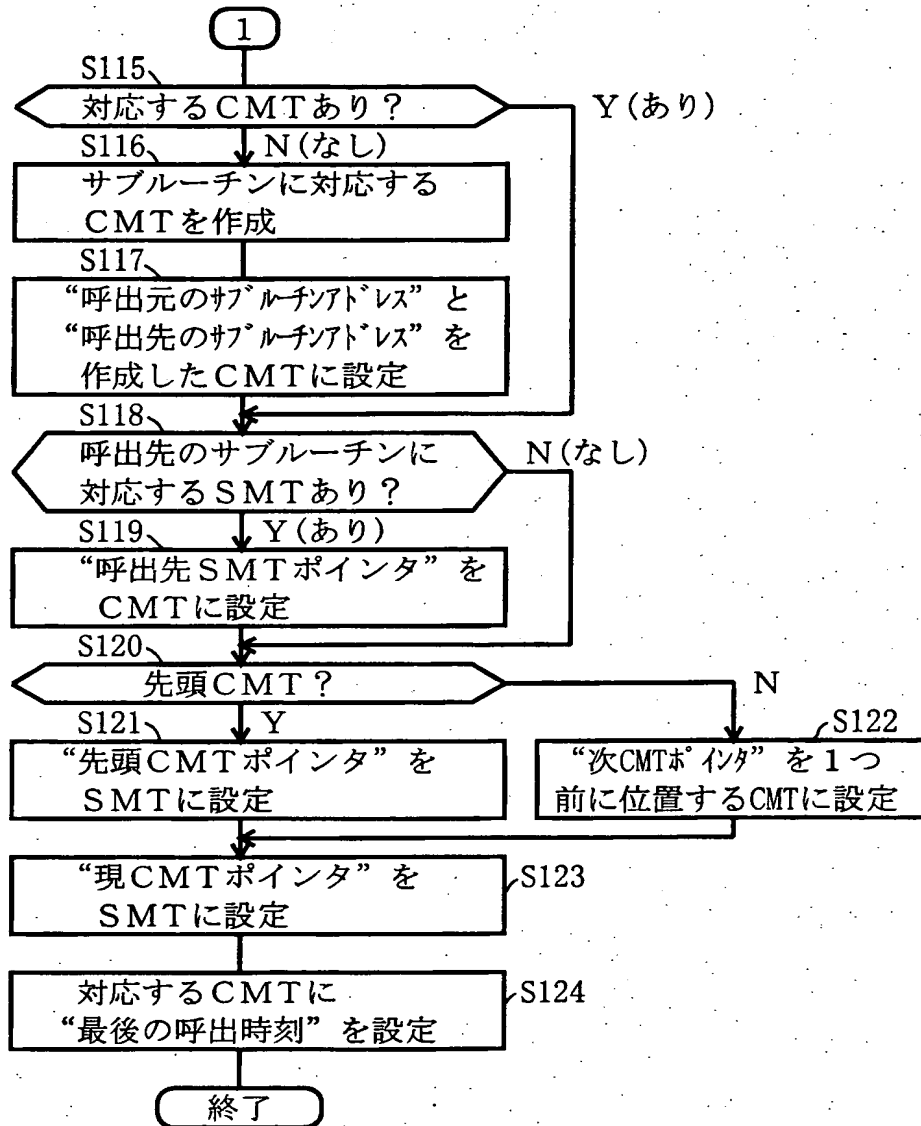
【図 5】

性能情報収集部の処理を示すフローチャート



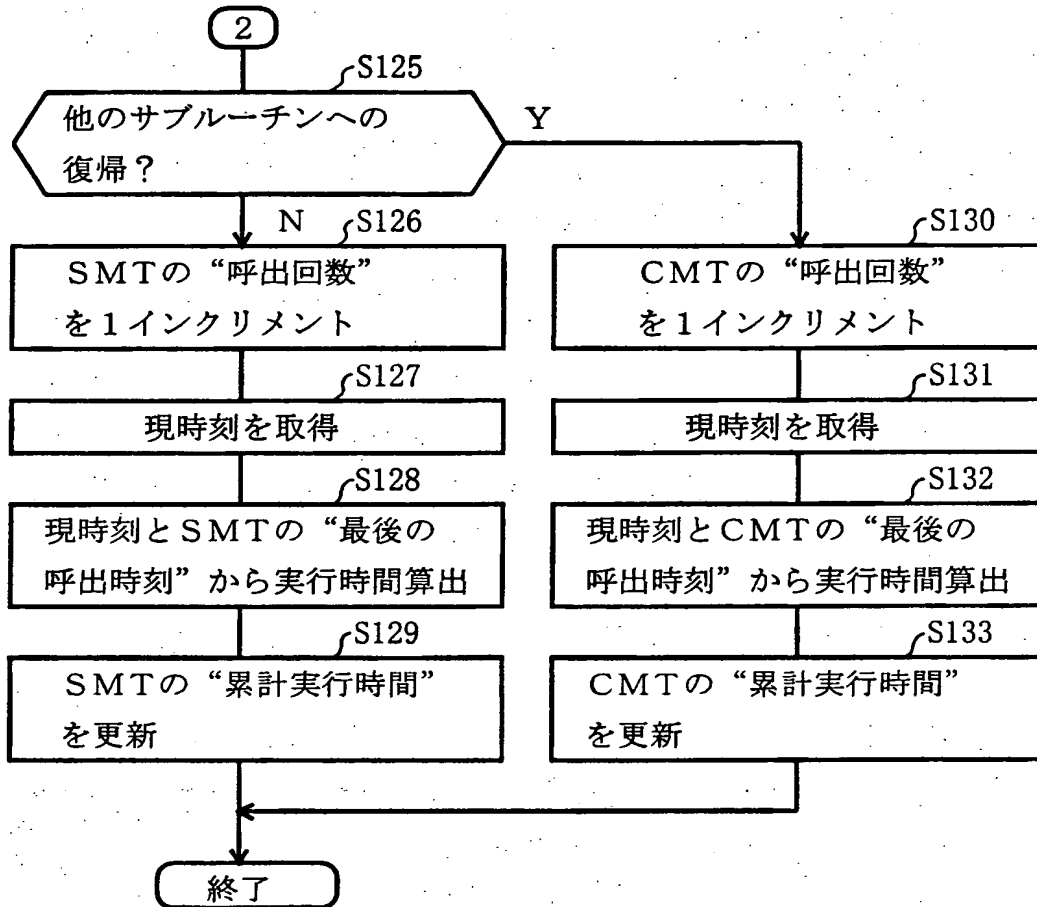
【図 6】

性能情報収集部の処理を示すフローチャート



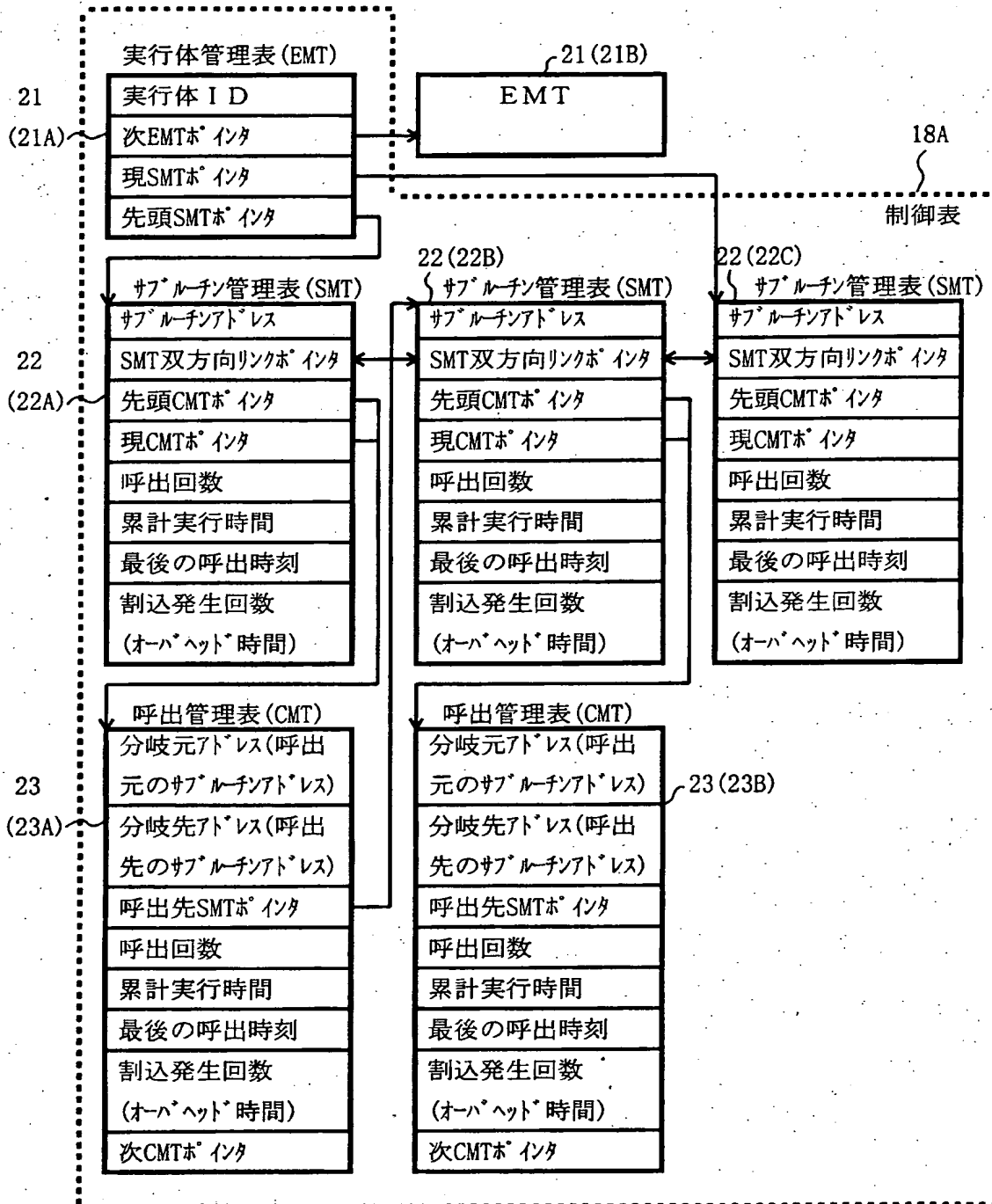
【図 7】

性能情報収集部の処理を示すフローチャート



【図 8】

制御表の構成例を示す図



【書類名】要約書

【要約】

【課題】プログラムに改変を加えることなくサブルーチンの性能情報を収集することができるプログラム性能情報収集装置を提供する。

【解決手段】プログラム性能情報収集装置 1 によると、プログラムの実行中に分岐命令が実行されると、CPU 2 による割込が発生する。この割込において、命令分析部 1 6 が、分岐命令がサブルーチンの実行に関する命令か否かを判別し、分岐命令が前記サブルーチンの実行に関する命令である場合には、性能情報収集部 1 7 が、サブルーチンの性能情報を取得して性能情報記憶部 1 8 に格納する。

【選択図】図 2

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中4丁目1番1号

氏 名 富士通株式会社